

le cnam

MPRO

RECHERCHE OPÉRATIONNELLE POUR LE DÉVELOPPEMENT DURABLE
RODD

PROTECTION DE LA BIODIVERSITÉ

Antoine Desjardins
Daphné Kany

Table des matières

1	Projet 1 - Sélection de réserves naturelles	2
1.1	Modélisation	2
1.2	Premières Résolutions	2
1.3	Passage à l'échelle	3
1.4	Modèle avancé	3
2	Projet 2 - Maîtrise des effets néfastes engendrés par la fragmentation du paysage	4
2.1	Premier Modèle	4
2.2	Modèle fractionnaire	4
2.3	Premières Résolutions	4
2.4	Passage à l'échelle	4
3	Projet 3 - Protection de la diversité génétique	4
3.1	Premier Modèle	4
3.2	Modèle Linéarisé	5
3.3	Premières Résolutions	5
3.4	Passage à l'échelle	5
4	Projet 4 - Exploitation durable de la forêt	6
4.1	Explication de P1	6
4.2	Rédaction de P2	6
4.3	Linéarisation de P2	6
4.4	Premières Résolutions	7
4.5	Ajout de contrainte	7
4.6	Passage à l'échelle	7
5	Projet 7 - Problème de planification culturelle durable	7
5.1	Représentation sous forme de graphe	7
5.2	Modélisation	8
5.3	Résolution	8
5.4	Formulation étendue	8
5.4.1	Génération de colonnes	8

1 Projet 1 - Sélection de réserves naturelles

1.1 Modélisation

La taille du côté de la parcelle est notée n (on n'a considéré que les parcelles carrées), et le nombre d'espèce est noté P . On introduit trois types de variables :

- $\forall (i, j) \in [1, n]^2$, une variable binaire $x_{i,j}$ qui vaut 1 si la zone (i, j) est sélectionnée (qu'elle soit tampon ou centrale)
- $\forall (i, j) \in [1, n]^2$, une variable binaire $y_{i,j}$ qui vaut 1 si la zone (i, j) est une zone centrale
- $\forall (k, i, j) \in [1, P] \cdot [1, n]^2$, une variable réelle $z_{k,i,j}$, qui nous permet de linéariser la contrainte sur la probabilité de survie en passant au logarithme. En fait, grâce aux contraintes, on aura $z_{k,i,j} = \log(1 - y_{i,j} * p_{k,i,j})$ pour les espèces en danger, et $z_{k,i,j} = \log(1 - x_{i,j} * p_{k,i,j})$ pour les espèces communes.

L'objectif est alors :

$$\min_x \sum_{i=1}^n \sum_{j=1}^n [x_{i,j} * c_{i,j}]$$

Et les contraintes linéarisées sont les suivantes :

- $y_{i,j} \leq x_{k,l}, \forall (i, j) \in [1, n]^2, \forall (k, l) \in \{i-1, i, i+1\} \cdot \{j-1, j, j+1\}$ (une zone est centrale ssi elle est entourée de zones tampon)
- $z_{k,i,j} \leq 0, \forall (k, i, j) \in [1, P] \cdot [1, n]^2$
- $z_{k,i,j} \geq \log(1 - p_{k,i,j} * y_{i,j}), \forall (k, i, j) \in [1, P] \cdot [1, n]^2$, k espèce en danger
- $z_{k,i,j} \geq \log(1 - p_{k,i,j} * x_{i,j}), \forall (k, i, j) \in [1, P] \cdot [1, n]^2$, k espèce commune
- $\sum_{i=1}^n \sum_{j=1}^n z_{k,i,j} \leq \log(1 - \alpha_k), \forall k \in [1, P]$ (probabilité de présence dans la réserve supérieure ou égale à α_k)

Expliquons un peu mieux l'obtention de cette dernière contrainte et la création des variables z .

On veut que la probabilité de survie de l'espèce k soit supérieure à α_k , ce qui se traduit par :

$1 - \prod_{i=1}^n \prod_{j=1}^n (1 - p_{k,i,j} * y_{i,j}) \geq \alpha_k$ (on présente ici le cas d'une espèce en danger. Pour traiter une espèce commune, il suffit de remplacer y par x)

Par passage au logarithme, on obtient :

$$\sum_{i=1}^n \sum_{j=1}^n \log(1 - p_{k,i,j} * y_{i,j}) \leq \log(1 - \alpha_k)$$

On introduit donc la variable $z_{k,i,j}$ pour linéariser la contrainte.

1.2 Premières Résolutions

Nous présentons ici les 4 premières instances proposées dans le sujet et leur résolution.

- cas 1 : $\alpha_k = 0.5 (k = 1, \dots, 6)$,
- cas 2 : $\alpha_k = 0.9 (k = 1, \dots, 3)$, et $\alpha_k = 0.5 (k = 4, \dots, 6)$,
- cas 3 : $\alpha_k = 0.5 (k = 1, \dots, 3)$, et $\alpha_k = 0.9 (k = 4, \dots, 6)$,
- cas 4 : $\alpha_k = 0.8 (k = 1, \dots, 3)$, et $\alpha_k = 0.6 (k = 4, \dots, 6)$.

cas	temps de calcul	nombre de noeuds	coût de la solution
1	0.06s	0	100
2	0.13s	0	271
3	0.13s	0	111
4	0.28s	0	198

TABLE 1 – Performances du modèle en fonction du paramètre α

cas	proba 1	proba 2	proba 3	proba 4	proba 5	proba 6
1	0.58	0.65	0.58	0.83872	0.76	0.6799
2	0.9153	0.913984	0.9244	0.98	0.892	0.9559
3	0.58	0.65	0.58	0.9	0.927	0.904
4	0.8236	0.832	0.82	0.977	0.784	0.7

TABLE 2 – Probabilités de survie des diverses espèces en fonction du paramètre α

1.3 Passage à l'échelle

Génération d'instances Nous avons créé un script **Julia** qui génère des instances aléatoires en fonction de certains paramètres. Les résultats qui suivent sont évalués sur ces instances. On prends en entrée P le nombre d'espèces et n la taille du côté du terrain (carré). On génère ensuite une grille de taille $n*n$ et l'on place les coûts entiers de manière mutuellement indépendante sur les cases suivant une loi uniforme entre 1 et 10. Puis, à chaque case, on associe de manière mutuellement indépendante une probabilité de survie pour l'espèce p selon une loi de Bernoulli de paramètre 0.9. La valeur de la probabilité de survie de l'espèce p dans la case i est tirée suivant une loi uniforme entre 0 et 0.5. Tout l'enjeu de la construction d'instances est de prendre garde à ne pas construire d'instances trop contraintes (sans solutions) ou trop peu contraintes (une solution est trouvée sans efforts par le système). Dans toute la suite, les instances générées sont décrites par leur paramètres (P, n) .

	p=6	p=10	p=15	p=20
n=10	2,17 t	2,64 t	3,37 t	4,05 t
n=15	4,72 t	5,77 t	7,43 t	8,6 t
n=20	9,98 t	10,5 t	13,8 t	15,4 t
n=25	13,8 t	16,4 t	20,9 t	25,7 t
n=30	19,6 t	24,5 t	31,3 t	36,1 t
n=40	39,1 t	47,6 t	57 t	66,9 t
n=50	65 t	78 t	95 t	112 t
n=100	440 t	492 t	556 t	626 t

TABLE 3 – Temps d'exécution en ticks en fonction des paramètres n et p

On remarque qu'à n constant, le temps d'exécution est à peu près linéaire par rapport à p (notamment, il double entre $p = 10$ et $p = 20$).

A p constant, le temps explose de façon exponentiel en n . Cela vient du fait que le nombre de case, et donc de contraintes, est en n^2 .

1.4 Modèle avancé

$$\min_{x, e} \sum_{i \in V} [x_i \cdot c_0 + \sum_{f \in F} x_{fi} \cdot c_{fi}]$$

$$x_i \geq x_{ikt}, \quad \forall i, \forall k, \forall t$$

On peut relaxer la contrainte de binarité sur les y et se contenter de poser que les y sont des réels entre 0 et 1. Formellement, on peut remplacer les contraintes

$$\forall i \in \{1, \dots, n\}, y_i \in \{0, 1\}$$

Par les contraintes

$$\forall i \in \{1, \dots, n\}, y_i \geq 0$$

2 Projet 2 - Maîtrise des effets néfastes engendrés par la fragmentation du paysage

2.1 Premier Modèle

On considère des zones carrées de côté n . On introduit deux types de variables booléennes :

- $\forall (i, j) \in [1, n]^2$, une variable binaire $x_{i,j}$ qui vaut 1 si la parcelle (i,j) est sélectionnée
- $\forall (i, j, k, l) \in [1, n]^4$, une variable binaire $z_{i,j,k,l}$ qui vaut 1 si les parcelles (i,j) et (k,l) sont sélectionnées, et que (k,l) est le plus proche voisin de (i,j) .

L'objectif est alors :

$$\min_{x,z} \frac{1}{\sum_{i,j=1}^n x_{i,j}} \sum_{i,j=1}^n \sum_{k,l=1}^n [z_{i,j,k,l} * d_{i,j,k,l}]$$

Et les contraintes sont les suivantes :

- $\sum_{i,j=1}^n x_{i,j} * c_{i,j} \leq B, \forall (i, j) \in [1, n]^2$ (contrainte de coût)
- $A_{min} \leq \sum_{i,j=1}^n x_{i,j} * a_{i,j} \leq A_{max}, \forall (i, j) \in [1, n]^2$ (contrainte d'aire)
- $z_{i,j,k,l} \leq x_{i,j} \forall (i, j) \in [1, n]^2, \forall (k, l) \in [1, n]^2$
- $z_{i,j,k,l} \leq x_{k,l} \forall (i, j) \in [1, n]^2, \forall (k, l) \in [1, n]^2$
- $\sum_{k,l=1}^n z_{i,j,k,l} \geq x_{i,j}, \forall (i, j) \in [1, n]^2$

2.2 Modèle fractionnaire

En posant $f(x, z) = \sum_{i,j=1}^n \sum_{k,l=1}^n [z_{i,j,k,l} * (-d_{i,j,k,l})]$ et $g(x, z) = \sum_{i,j=1}^n x_{i,j}$, alors on cherche à maximiser $\frac{f(x,z)}{g(x,z)}$, avec $X = \{(x, z) \text{ respectant les contraintes décrites plus haut}\}$.

2.3 Premières Résolutions

Nous présentons ici les 3 premières instances proposées dans le sujet et leur résolution.

- cas 1 : $A_{max} = 30, A_{min} = 35, B = 920$
- cas 2 : $A_{max} = 20, A_{min} = 21, B = 520$
- cas 3 : $A_{max} = 70, A_{min} = 75, B = 3500$

2.4 Passage à l'échelle

Nous avons créé un script **Julia** qui génère des instances aléatoires en fonction de certains paramètres. Les résultats qui suivent sont évalués sur ces instances. On prends en entrée m la largeur du terrain et n la longueur du terrain. On génère ensuite une grille de taille $n * m$ et l'on place les coûts entiers de manière mutuellement indépendante sur les cases suivant une loi uniforme entre 1 et 10.

Voilà les temps de calculs obtenus selon la taille de n et m :

n=m=10	n=m=15	n=m=20	n=m=30	n=m=50
2s	8s	58s	5min	>10min

La complexité du problème est bien exponentielle en la taille du coté de la grille ($n*m$ cases), ce qu'on retrouve bien avec ces temps de calcul. Le problème est très rapide à résoudre pour des instances de taille <20 , et devient vite trop lourd pour les grilles de côté supérieur à 30.

3 Projet 3 - Protection de la diversité génétique

3.1 Premier Modèle

Le nombre d'allèles par gène est considéré comme étant le même pour chaque gène et ce nombre est noté J . Le nombre de parents est noté P , avec M l'ensemble des mâles et F l'ensemble des femelles. On introduit une variable :

- $\forall i \in [1, P]$, une variable $x_i \in \mathbb{N}$ qui représente le nombre d'enfants du parent i .

On introduit également les données du problème sous la forme :

- $\forall (i, j) \in [1, P] * [1, J]$, une $y_{i,j} \in \mathbb{N}$ qui représente le nombre de chromosomes exprimant l'allèle j chez le parent i . $y_{i,j}$ prends des valeurs allant de 0 à $2*n$ avec n le nombre de paires de chromosomes par individu.
- $f_{i,j} = 1 - \frac{y_{i,j}}{2} \forall (i, j) \in [1, P] * [1, J]$

L'objectif est alors :

$$\min_x \sum_{j=1}^J \prod_{i=1}^P f_{i,j}^{x_i}$$

Et les contraintes sont les suivantes :

- $\sum_{i \in M} x_i = P$
- $\sum_{i \in F} x_i = P$

Ce problème n'est pas linéaire. Nous allons donc essayer de linéariser ce problème par la méthode du logarithme approché.

3.2 Modèle Linéarisé

On réécrit notre modèle afin de le linéariser. On introduit les variables z_j et t_j pour chaque allèle j . On pose $I_{j,1}$ =[parents hétérozygotes pour le gène associé à l'allèle j], et $I_{j,2}$ =[parents homozygotes pour le gène associé à l'allèle j]. L'objectif devient :

$$\min_x \sum_{j=1}^J z_j$$

sous contraintes :

- $\sum_{i \in M} x_i = P$
- $\sum_{i \in F} x_i = P$
- $z_j \geq t_j - \sum_{i \in I_{j,2}} x_i, \forall j \in [1, J]$
- $z_j \geq 0, \forall j \in [1, J]$
- $\log(t_j) \geq \sum_{i \in I_{j,1}} \log(f_{i,j}) * x_i, \forall j \in [1, J]$

3.3 Premières Résolutions

Nous présentons ici la première instance proposée dans le sujet et sa résolution.

- cas 1 : on limite à 3 le nombre d'enfants par individu
- cas 2 : on limite à 2 le nombre d'enfants par individu

cas	temps de résolution	nbr noeuds	proba par allèle	\mathbb{E} [nbr allèles disparus]	BI
$x \leq 3$	0.36 ticks	0	0 sauf pour l'allèle 4, où p=1	1	1
$x \leq 2$	0.32 ticks	0	0 sauf pour l'allèle 4, où p=1	1	1

3.4 Passage à l'échelle

Nous avons créé un script `Julia` qui génère des instances aléatoires en fonction de certains paramètres. Les résultats qui suivent sont évalués sur ces instances. Cependant, dans la plupart des cas, les instances générées n'admettaient pas de solution. Nous n'avons donc pas traité cette partie.

4 Projet 4 - Exploitation durable de la forêt

4.1 Explication de P1

On veut maximiser la somme des populations des espèces e_1 et e_2 .
 Pour l'espèce e_1 , la population vaut $\sum_{i=1}^N \sum_{j=1}^M t_{i,j}(1 - x_{i,j})$, puisque qu'on compte $t_{i,j}$ à condition que la parcelle $s_{i,j}$ soit coupée, donc à condition que $x_{i,j}$ vaille 0.
 Pour l'espèce e_2 , on veut compter toutes les lisières séparant une parcelle coupée et une parcelle non coupée. Donc, pour éviter de compter deux fois les lisières en parcourant toute la matrice, on compte uniquement les bordures des parcelles non coupées.
 Si la parcelle est non coupée, $d_{i,j}$ vaut le nombre de voisins de $s_{i,j}$ qui sont non coupés. ($d_{i,j} = \sum_{(k,l) \in A_{i,j}} x_{k,l}$). Il faut donc soustraire ce nombre au nombre total de voisins de $s_{i,j}$ qui vaut 4.
 Si la parcelle est coupée, on veut que $\sum_{(i,j) \in M*N} x_{k,l} 4 * x_{i,j} - d_{i,j} = 0$. Or ici cette somme est égale à $\sum_{(i,j) \in M*N} d_{i,j}$. La contrainte sur $d_{i,j}$ impose $d_{i,j} \geq \sum_{(k,l) \in A_{i,j}} x_{k,l} - |A_{i,j}|$, donc $d_{i,j} \geq 0$ puisque $\sum_{(k,l) \in A_{i,j}} x_{k,l} \leq |A_{i,j}|$.
 Ainsi, dans tous les cas, le nombre de lisière vaut $\sum_{(i,j) \in M*N} x_{k,l} 4 * x_{i,j} - d_{i,j}$.
 L'objectif du problème peut donc bien s'écrire

$$\max_x w_1 \sum_{i,j \in M*N} t_{i,j} * (1 - x_{i,j}) + w_2 lg \sum_{i,j \in M*N} 4x_{i,j} - d_{i,j}$$

Et les contraintes :

- $d_{i,j} \geq \sum_{(k,l) \in A_{i,j}} x_{k,l} - |A_{i,j}|(1 - x_{i,j}), \forall (i,j) \in M * N$
- $d \geq 0$
- $x \in [0, 1]^{M*N}$

4.2 Rédaction de P2

On veut à présent modéliser le problème de façon quadratique. On garde les variables binaires $x_{i,j}$ définies comme pour P1, mais on les étend sur $(M+1)*(N+1)$ pour faciliter la gestion des bords. On garde aussi les voisinages $A_{i,j}, \forall (i,j) \in [M * N]$.
 Pour l'espèce 2, on compte les bordures des parcelles non coupées.
 L'objectif est donc

$$\max_x w_1 \sum_{i,j \in M*N} t_{i,j} * (1 - x_{i,j}) + w_2 lg \sum_{i,j \in M*N} \sum_{k,l \in A_{i,j}} x_{i,j}(1 - x_{k,l})$$

s.t

- $x \in [0, 1]^{(M+1)*(N+1)}$

4.3 Linéarisation de P2

On cherche à présent à linéariser le problème P2. On introduit, $\forall (i,j) \in (M * N), \forall (k,l) \in A_{i,j}$ la variable continue $z_{i,j,k,l}$, qui, grâce aux contraintes, vaut $x_{i,j}(1 - x_{k,l})$. Le nouveau problème s'écrit :

$$\max_x w_1 \sum_{i,j \in M*N} t_{i,j} * (1 - x_{i,j}) + w_2 lg \sum_{i,j \in M*N} \sum_{k,l \in A_{i,j}} z_{i,j,k,l}$$

s.t

- $z_{i,j,k,l} \geq x_{i,j}, \forall (i,j) \in (M * N), \forall (k,l) \in A_{i,j}$
- $z_{i,j,k,l} \geq 1 - x_{k,l}, \forall (i,j) \in (M * N), \forall (k,l) \in A_{i,j}$
- $x \in [0, 1]^{(M+1)*(N+1)}$

Montrons que la matrice des contraintes est bien TU.

En fait, on est face à une matrice constituée de la juxtaposition de matrices I ou -I. Or une propriété est que la juxtaposition d'une matrice TU et de l'identité est TU. Donc la juxtaposition de matrices identités est bien TU. De plus, le membre de l'inégalité est entier. On peut donc résoudre la relaxation continue de notre problème.

4.4 Premières Résolutions

Nous présentons ici la première instance proposée dans le sujet et sa résolution.

- cas 1 : $w_1 = 1$, $w_2 = 5$
- cas 2 : $w_1 = 2$, $w_2 = 1$

		temps (s)	# noeuds
cas 1	P1	0.11	0
cas 1	P2	0.02	0
cas 2	P1	0.03	0
cas 2	P2	0.03	0

4.5 Ajout de contrainte

On ajoute la contrainte que la somme des $x \in [0, 1]^{(M+1)*(N+1)}$ est majoré par 60. Cela revient à ajouter une ligne de 1 en bas de notre matrice des contraintes. Prise seule, la matrice constituée de cette ligne est TU. On lui juxtapose ensuite des matrices identité, donc la matrice finale est bien TU.

		temps (s)	# noeuds
cas 1	P1	0.22	0
cas 1	P2	0.02	0
cas 2	P1	0.06	0
cas 2	P2	0.00	0

4.6 Passage à l'échelle

Nous avons créé un script `Julia` qui génère des instances aléatoires en fonction de certains paramètres. Les résultats qui suivent sont évalués sur ces instances. On prends en entrée la taille de la grille et l'on génère les t_{ij} aléatoires avec des valeurs tirées uniformément entre 70 et 100.

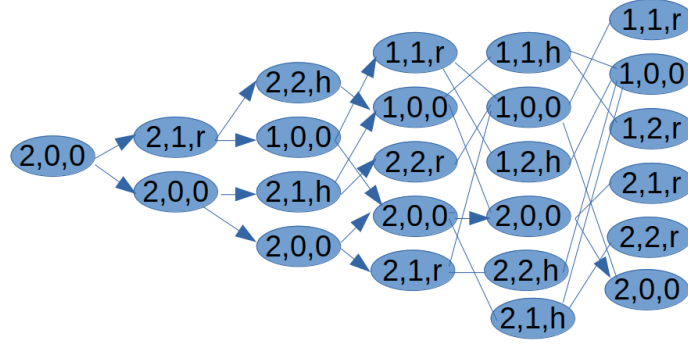
taille		temps (s)	# noeuds
10*10	P1	0.22	0
10*10	P2	0.02	0
15*15	P1	0.06	0
15*15	P2	50.29	0
20*20	P1	0.06	0
20*20	P2	448	0
30*30	P1	0.13	0
30*30	P2	intractable	0

Nous observons que P1 subit fort bien le passage à l'échelle, tandis que P2 a très vite du mal. On peut en effet remarquer que P2 a un nombre de variables + contraintes en puissance de 4, ce qui explique cette explosion du temps de calcul.

5 Projet 7 - Problème de planification culturelle durable

5.1 Représentation sous forme de graphe

En étudiant un peu le problème, on se rend compte qu'à chaque saut de temps, n parcelle passe d'un état à un autre selon différentes règles qui peuvent se résumer à un chemin orienté dans un graphe, chaque arc passant d'un état au temps t à un autre état au temps $t + 1$.



5.2 Modélisation

En généralisant l'observation du dessus, on modélise le PLNE qui suit. On introduit les variables binaires $x_{u,t,p}$, qui vaut 1 ssi la parcelle p utilise le chemin u du graphe au temps t . L'objectif (minimiser le nombre de parcelles utilisées) est alors :

$$\min \sum_{p \in P} \sum_{u \in \delta^+(v_0)} x_{u,1,p}$$

Et les contraintes sont les suivantes :

- $\sum_{u \in \delta^+(v_{p,t})} x_{p,t,u} = \sum_{u \in \delta^-(v_{p,t})} x_{p,t,u}, \forall t \in [1, T], \forall p \in [1, P]$ (conservation du flot)
- $D_{j,t} \leq \sum_{p \in P} \sum_{u \in \delta^+(v_{t,p,j})} x_{u,t,p} * R(u) \forall j \in [1, J], \forall t \in [1, T]$ (satisfaction de la demande en chaque culture à chaque semestre)

5.3 Résolution

Nous obtenons une solution optimale en cultivant 19 parcelles en rotation en seulement 0.58 secondes pour 0 noeuds

5.4 Formulation étendue

Avec cette nouvelle formulation, le pb s'écrit comme suit (on considère que x_0 compte le nombre de parcelles non utilisées, R l'ensemble des rotation possibles) :

$$\max x_0$$

sc :

- $x_0 + \sum_{r \in R} x_r = P$ (une rotation par parcelle)
- $D_{j,t} \leq \sum_{r \in R} x_r * R(j, t, r) \forall j \in [1, J], \forall t \in [1, T]$ où $R(j, t, r)$ est le rendement de l'espèce j au temps t sur le scénario r

5.4.1 Génération de colonnes